JSON WITH PYTHON

This tutorial will teach you how to encode and decode JSON objects using Python programming language. Let's start with preparing environment to start our programming with Python for JSON.

Environment

Before you start with encoding and decoding JSON using Python, you will need to install any of the JSON modules available. For this tutorial I downloaded and installed Demjson as follows:

```
$tar xvfz demjson-1.6.tar.gz
$cd demjson-1.6
$python setup.py install
```

JSON Functions

Function	Libraries
encode	Encodes the Python object into a JSON string representation.
decode	Decodes a JSON-endoded string into a Python object

Encoding JSON in Python (encode)

Python encode() function encodes the Python object into a JSON string representation.

Syntax:

```
demjson.encode(self, obj, nest_level=0)
```

Example

The following example shows arrays under JSON with Python

```
#!/usr/bin/python
import demjson

data = [ { 'a' : 1, 'b' : 2, 'c' : 3, 'd' : 4, 'e' : 5 } ]

json = demjson.encode(data)
print json
```

While executing, this will produce following result:

```
[{"a":1,"b":2,"c":3,"d":4,"e":5}]
```

Decoding JSON in Python (decode)

Python can use demjson.decode() function for decoding JSON. This function returns the value decoded from json to appropriate Python type.

Syntax:

```
demjson.decode(self, txt)
```

Example

The following example shows how Python can be used to decode JSON objects.

```
#!/usr/bin/python
import demjson

json = '{"a":1,"b":2,"c":3,"d":4,"e":5}';

text = demjson.decode(json)
print  text
```

While executing, this will produce following result:

```
{u'a': 1, u'c': 3, u'b': 2, u'e': 5, u'd': 4}
```